Exploiting Recurring Structure in a Semantic Network

Shawn R. Wolfe, Richard M. Keller

Computational Sciences Division, MS 269-2,
NASA Ames Research Center, Moffett Field, CA USA 94035
{Shawn.R.Wolfe, Richard.M.Keller}@nasa.gov

Abstract. With the growing popularity of the Semantic Web, an increasing amount of information is becoming available in machine interpretable, semantically structured networks. Within these semantic networks are recurring structures that could be mined by existing or novel knowledge discovery methods. The mining of these semantic structures represents an interesting area that focuses on mining both *for* and *from* the Semantic Web, with surprising applicability to problems confronting the developers of Semantic Web applications. In this paper, we present representative examples of recurring structures and show how these structures could be used to increase the utility of a semantic repository deployed at NASA.

1 Introduction

The Semantic Web effort, with its emphasis on machine interpretable information, is creating exciting new research possibilities in knowledge discovery. Primarily, this research has focused on adapting known techniques to the Semantic Web, either by mining conventional information sources to augment a semantic network, or by extracting information from a semantic network that is then mined for conventional purposes. The overlap of these areas is an entirely new arena for knowledge discovery: mining the semantic network to enhance the semantic network itself or aid in its use. We are interested in a particular mining problem where both the input and output is a semantic network, namely finding recurring, similar semantic structures in a larger semantic network. A number of Semantic Web applications store information in large networks, notably ODESeW [1], OntoWebber [2], SEAL [3], OntoWeb [4], the KAON suite [5], BrainEKP [6], Semagix Freedom [7] and our own SemanticOrganizer [8]. Our focus in this paper is not an algorithm for discovering recurring semantic structure, but rather how such structures can be used once identified, namely for:

- **Enforcing consistency with rules**. Identifying all the rules needed to enforce logical consistency in a semantic network is a non-trivial task. Patterns of recurring structures can be used to generate candidate rules.
- **Aiding in analysis**. The semantic organization of information makes information easier to find, but the analysis process is still manual. Identifying recurring structures in the semantic network would automate part of the analysis process.

- **Performing ontology maintenance**. Ontology maintenance for persistent and evolving Semantic Web applications is time consuming and difficult, resulting in less than optimal modeling decisions that impact the usability of the system. The identification of recurring structures can indicate patterns that suggest useful changes to the ontology.
- **Reducing network complexity**. As the size of a semantic network grows, it becomes increasingly difficult to navigate or display the information space. Abstracting recurring structures that match the same pattern can reduce the size and complexity of the network representation, making it more manageable with existing navigation and visualization techniques.

2 Exploiting Recurring Semantic Structure

We use semantic templates [9] to define recurring semantic structure. A semantic template consists of a set of abstracted RDF-like triples, and the matches to this template are the recurring semantic structure in the network. Figure 1 gives an example of an abstract graph pattern in an RDQL format and a matching set of statements. In the following subsections, we describe various ways in which recurring semantic structure can be exploited to improve the utility of systems that use semantic networks.

```
(?x researcher-in ?y)
(?x authored ?z)
(?z submitted-to ?w)
(?w has-topic-area ?y)
```

matches

```
("Shawn Wolfe" researcher-in "Semantic Web")
("Shawn Wolfe" authored "Exploiting Recurring Structure…")
("Exploiting Recurring Structure…" submitted-to "SW Mining Workshop")
("SW mining Workshop" has-topic-area "Semantic Web")
```

Figure 1. Example of a semantic template and a corresponding match in the semantic network.

2.1 Enforcing consistency with rules

Users of a semantically structured repository cannot be expected to create every relevant link between nodes. However, failure to create all such links leads to a less complete, less accurate and subsequently less useful network. We feel that it is necessary to augment the semantic network by providing additional links through inference. Some of the supporting inference rules can be derived from the structure of the ontology (e.g., deriving a property from a sub-property), whereas other rules are based on domain knowledge. Figure 2 gives an example of a rule based on domain knowledge, stating that samples gathered during an experiment must be collected at

the site of that experiment. It is these domain-specific rules that we seek to discover through the identification of recurring semantic structure.

We regard inference rules as composed of semantic templates, with the antecedent and consequent sections each consisting of a semantic template. Assuming a relatively complete and representative semantic network, it should be possible identify possible domain-specific inference rules by finding a significant number of matches to candidate antecedents and consequents. Even a fairly unsophisticated technique that generates a large number of undesirable candidate rules would be helpful, since identifying correct rules from a large set of candidates is easier than deriving them through manual domain analysis.

```
(?sample gathered-during ?experiment)
(?experiment conducted-at ?site)
->
(?sample collected-from ?site)
```

Figure 2. Example of an inference rule from a biology domain.

2.2 Aiding in analysis

Recurring structures can also reveal interesting features in the semantic network. For example, consider a semantic network modeling a biological experiment measuring the effects of salinity and pH level on stored cultures. An algorithm that generates candidate inference rules by identifying recurring structure could generate the rules in Figures 2-3. However, the rule in Figure 3 would reveal a result of the experiment, thus aiding the biologist in analyzing the results. The difference between these two candidate rules is that the rule in Figure 2 would be used to enforce semantic consistency, whereas the rule in Figure 3 reveals something interesting about the domain.

```
(?culture salinity "high")
(?culture pH-level "9.0")
->
(?culture exhibits "speckling")
```

Figure 3. Example of an unexpected rule that reveals a previously unknown correlation

Statistical analysis on the recurring semantic structure can also reveal interesting features in a semantic network. Consider a semantic network for an investigation domain that has information on 1000 total mishaps. Figures 4-7 show three semantic templates for this domain and the number of matches for each. Since one out of ten mishaps involves a jackscrew in this example, we would have expected only four or so MD-80 mishaps to involve jackscrews. Since this number is significantly higher, an investigator may deduce that there is an issue with reliability of jackscrews in MD-80 airplanes.

```
(?mishap involves ?plane)
(?plane model "MD-80")
```

Figure 4. A semantic template that has 40 matches.

```
(?mishap involves ?plane)
(?mishap concerns "jackscrew-failure")
```

Figure 5. A semantic template that has 100 matches.

```
(?mishap involves ?plane)
(?plane model "MD-80")
(?mishap concerns "jackscrew-failure")
```

Figure 6. A semantic template that has 16 matches, indicating a correlation between jackscrew failures and MD-80 mishaps.

2.3 Performing ontology maintenance

The identification of recurring structure can also be benefit ontology development. In our experience, ontologies require significant maintenance as application requirements change over time. The identification of recurring semantic structure can suggest approaches to revising an existing ontology based on this evolving pattern of usage. One form of ontology change supported by semantic template identification is specialization, where a single concept in an ontology is elaborated by adding several more specific subconcepts beneath the original, thus providing for more accurate and therefore more meaningful modeling.

Consider the patterns described in Figures 7-9 from a project management ontology. Three different subconcepts of document are suggested by the documents that would match these templates: a submitted publication concept, an experimental procedure concept, and software documentation concept. Additional analysis of the recurring structure could reveal that no document matches more than one of these patterns: after all, software documentation is not submitted to conferences, experiment procedures do not describe software, and so on. Such realizations may suggest to the ontology maintainer that the document concept should be split into several subconcepts: publications, experimental procedures and software documentation. This specialization would lead to a more constrained domain model that prevents some illogical pairings (such as a given document describing software and following an experimental protocol), and indeed manual analysis lead us to a similar specialization in our ontology.

```
(?document submitted-to ?conference)
(?document acceptance-status ?status)
```

Figure 7. A publication document template.

```
(?document tests ?hypothesis)
(?document follows ?experimental-protocol)
```

Figure 8. An experiment procedure template.

```
(?document describes ?software-module)
(?document has-version ?software-version)
```

Figure 9. A software documentation template.

2.4 Reducing network complexity

Finally, repeating patterns can serve as an aid to visualization and navigation. We have found that our semantic networks have quickly grown to the point where people have trouble navigating them [10]. A display of the immediate neighborhood of a semantic node is often insufficient context for users, but displaying the entire network is infeasible due to the large number of nodes and edges. One approach to solve this problem is to combine similar nodes into a composite node, thus reducing the complexity of the space and making it possible to visualize with conventional techniques.

Figure 10 presents a semantic template from a biological domain. In this domain, scientists perform experiments collecting measurements on samples. Any set of measurements that match the template with the same values for `?experiment`, `?date`, and `?sample` would be indistinguishable with respect to this template, thereby forming an equivalence class. We envision developing a technique, either by explicitly choosing important and unimportant differences or through some implicit analysis, which would allow us to collapse such similar nodes in appropriate situations, as illustrated in Figure 11.

```
(?experiment produces ?measurement)
(?measurement collected-on ?date)
(?measurement measures ?sample)
```

Figure 10. A template defining an equivalence class.

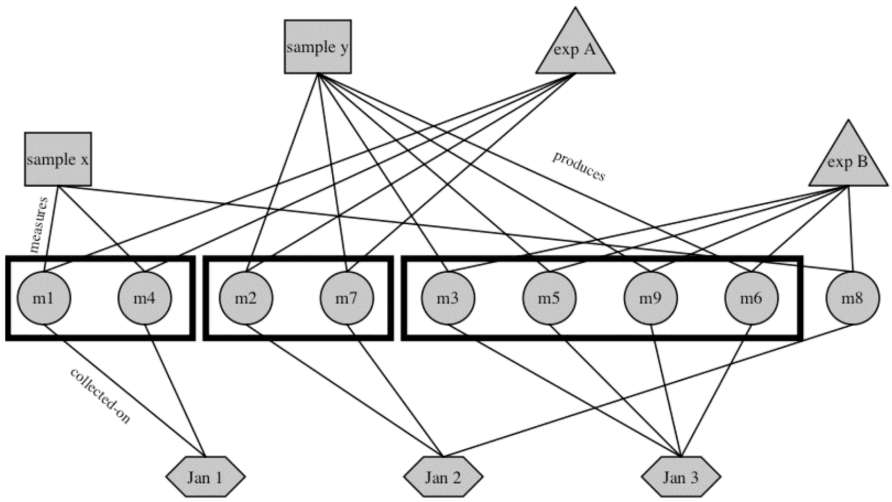

Figure 11. A semantic network with boxes around nodes that could be combined into composite nodes according to the semantic template given in Figure 10.

3 Conclusion

We have presented a simple definition of recurring semantic structure and discussed several ways in which it could used to improve a repository that stores information in a semantic network. Our analysis has led us to advocate mining for recurring semantic structure as a fruitful area of research: the problem lies in an area relatively unexplored and the simple definition of semantic structure should be amenable to straightforward knowledge discovery methods. Furthermore, even unsophisticated techniques could be beneficial, as relatively inaccurate and imprecise results still offer some automated assistance where there currently is none.

4 Acknowledgements

We would like to thank the ScienceDesk team and Deepak Kulkarni for their contributions to this paper. Our work on SemanticOrganizer is funded by the NASA Intelligent Systems Project of the Computing, Information, and Communications Technology Program.

5 References

1. O. Corcho, A. Gomez-Perez, A. Lopez-Cima, V. Lopez-Garcia, and M. Suarez-Figueroa, "ODESeW. Automatic generation of knowledge portals for Intranets and Extranets," The Semantic Web - ISWC 2003, vol. LNCS 2870, pp. 802-817, 2003.
2. Y. Jin, S. Xu, S. Decker, and G. Wiederhold, "OntoWebber: a novel approach for managing data on the Web," International Conference on Data Engineering, 2002.
3. N. Stojanovic, A. Maedche, S. Staab, R. Studer, and Y. Sure, "SEAL - a framework for developing semantic portals," Proceedings of the International Conference on Knowledge capture, pp. 155-162, 2001.
4. P. Spyns, D. Oberle, R. Volz, J. Zheng, M. Jarrar, Y. Sure, R. Studer, and R. Meersman, "OntoWeb - a semantic Web community portal," Fourth International Conference on Practical Aspects of Knowledge Management, 2002.
5. E. Bozsak, M. Ehrig, S. Handschuh, A. Hotho, A. Maedche, B. Motik, D. Oberle, C. Schmitz, S. Staab, L. Stojanovic, N. Stojanovic, R. Studer, G. Stumme, Y. Sure, J. Tane, R. Volz, and V. Zacharias, "KAON-towards a large scale Semantic Web," Proceedings of EC-Web, 2002.
6. "BrainEKP." Santa Monica, CA: TheBrain Technologies Corporation, 2004, http://www.thebrain.com.
7. A. P. Sheth and C. Ramakrishnan, "Semantic (Web) Technology In Action: Ontology Driven Information Systems for Search, Integration and Analysis," IEEE Data Engineering Bulletin, vol. 26, pp. 40-48, 2003.
8. R. M. Keller, D. C. Berrios, R. E. Carvalho, D. R. Hall, S. J. Rich, I. B. Sturken, K. J. Swanson, and S. R. Wolfe, "SemanticOrganizer: A Customizable Semantic Repository for Distributed NASA Project Teams," *submitted to*: ISWC-2004, 2004.
9. D. C. Berrios, S. R. Wolfe, and R. M. Keller, "A Common Approach to Searching and Extending a Semantic Web," *submitted to*: ISWC-2004, 2004.
10. R. M. Keller, and D. R. Hall, "Developing Visualization Techniques for Semantics-based Information Networks," Workshop on Visualization in Knowledge Engineering, 2nd International Conference on Knowledge Capture October.